CybOnLine - User Manual

On-line fuzzy modelling for process control applications

Published and distributed by Process Cybernetics Limited

Copyright ©1998 Process Cybernetics Limited
All Rights Reserved

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photographic, recording, or otherwise, without the prior written permission of Process Cybernetics Ltd.

Disclaimer

Process Cybernetics Limited makes no representation or warranties with respect to the programs or manual and specifically disclaim any expressed or implied warranties of merchantability or fitness for purpose. In no event shall Process Cybernetics Ltd be held liable for direct, indirect, consequential or incidental damages resulting from any defect or omission in the manual, or other related items or processes, including, but not limited to , any interruption of service, loss of business, income, anticipated profit, use, information, or any other consequential damages.

Trademarks

Microsoft and Windows are registered trademarks of the Microsoft Corporation.

Table of Contents

Introduction 1
System Requirements 2
Installing CybOnLine 2
Verifying CybOnLine 2

Building Models for use in CybOnLine 4
Building Inferential Models 4
Building 'What-If' models 7

Linking to a SCADA System 12
Dynamic Data Exchange (DDE) 12
Triggering the calculation 14
Automatic triggering 14
External Triggering 15
Running CybOnLine 16
Adding CybOnLine to the Windows 'StartUp' Folder 16
Adding CybOnLine to the SCADA software's start-up list 18
The CybOnLine Faceplate and Linking to the SCADA system. 19

The 'Outputs' card 20
Comms/Triggers 22
Inputs 23
SCADA connections 24
Linking 'what-if' models 26

Other Services from Process Cybernetics . . 29
Other products 30
CybMod - A fuzzy modelling program for process control 30
CybFIMC - Model-based fuzzy control 31
Services from Process Cybernetics 32
Training 32
Consultancy 32
Software services 33
Contacting us 34

0

Introduction

Welcome to CybOnLine. CybOnLine offers a flexible and easy to use way of integrating fuzzy modelling technology with your existing Windows based SCADA system. CybOnLine can be used to provide inferred measurements (where a difficult to measure quantity is inferred from other measurements), or 'what-if' modelling to support process operators.

CybOnLine uses models which have been developed in CybMod, Process Cybernetics' fuzzy modelling package. The models are loaded at the command line as CybOnLine starts. CybOnLine then acts as a DDE server allowing DDE capable SCADA software to link to it simply by setting a few I/O addresses in the SCADA package. CybOnLine runs in the background carrying out the model calculations, and passing the information to the SCADA package, allowing you to easily integrate the fuzzy modelling results into your control schemes.

We have developed CybOnLine from leading edge research in fuzzy modelling and control in the process industries. Process engineers have designed the package with the process industries in mind. We are continually developing our products, and would be delighted to receive your feedback.

System Requirements

Pentium PC running Microsoft Windows 95, 98, or NT. A minimum of 16 MB of RAM is recommended.

Installing CybOnLine

Insert the diskette number 1 into the floppy disc drive. Choose the **Run** command from the **Start** menu, and in the Run window type:

a:\setup.exe

and then press enter.

Follow the on-screen instructions to complete the installation.

Verifying CybOnLine

The first time you run CybOnLine you will be taken to a licence verification dialogue. Simply enter your password and press the 'OK' button, then follow the on-screen instructions.

Each copy of CybOnLine is licensed to run on only a single machine. The number of installations is strictly limited by the protection software, and these installations are only available to help you in the event of a hardware problem. If you exceed the number of permitted installations you should contact Process Cybernetics. Process Cybernetics reserves the right to charge for, or refuse to supply, additional registrations.

1

Building Models for use in CybOnLine

The two main uses for CybOnLine are for inferential and 'what-if' modelling. Although the models in both cases will be built using our modelling package CybMod, there are slight differences in the model building process.

Building Inferential Models

An inferential model is used to estimate the value of a difficult to measure variable from the values of other easy to measure variables. The most commonly inferred variables are compositions (e.g. estimating the mole fraction on a distillation column plate from temperature and pressure data).

A feature that is common with most inferential systems is that they don't involve dynamics. The measurements used to infer the 'difficult' value are related to it in a purely functional way that involves neither differential equations nor time delays.

The data used for inferential modelling can come from two sources. The required data may sometimes be available in tables of data (e.g. boiling point data), and in this case the fuzzy model simply serves as a convenient way to transform this tabular data to a continuous form. More commonly the data is available from historical plant data. Difficult to measure quantities are often obtained from routine sampling and analysis. Historical records of these analytical results can be merged with the recordings of the other data and used to provide a data set for modelling purposes.

To demonstrate the procedure involved in generating an inferential model a sample data set is included with this package. You can find the data in the last disk in the CybOnLine distribution set under the filename 'press.dat'. The data consists of several samples each of three fields of data which represent, respectively, the temperature, the pressure and the mass of gas contained within a vessel. The data were generated from an ideal gas type equation, i.e.

$$Mass \text{ ' } 2.646.10^{\&3} \frac{Pressure}{Temperature}$$

This data was generated purely for tutorial purposes. In a real situation, if you had an equation of this sort it would, of course, be much more sensible to include it directly in the SCADA systems rather than building a fuzzy model.

For the tutorial let's build a model which infers the mass of gas in the vessel from the temperature and pressure. First of all load the data into CybMod (see the CybMod manual for instructions). Change the field names as follows: 'Field_0' to 'Temperature'; 'Field_1' to 'Pressure'; and 'Field_2' to 'Mass'.

Unlike data from a dynamic system, the samples in this data file are in no particular sequence. The samples are completely unrelated to time, and this means that the cross-correlation results are mostly meaningless. The correlation coefficient at zero lag is useful, however, as it represents the correlation between the various fields in the data with no dynamic lag between them.

Create a model by selecting 'Mass' as the output and pressing the **Create** button on the main window dialogue. A card containing the new model will appear. Note that the model currently has a single input; 'Mass' at a lag of one. We don't want to create a dynamic model, so the first thing we need to do is change this lagged output into a model input that we do want. Double click on the lagged input, change the variable to 'Temperature',

and change the lag to zero.

Now add the other input we want on the model, the 'Pressure' at zero lag. Identify the model using sequential presentation and all the data. Save the model as "c:\program files\cybonline\tut.pcg" for use in the tutorial in the next chapter.

Care has to be taken when interpreting the results of the model analysis tools, which have been primarily designed for dynamic modelling. The prediction plot will be fine for point-to-point comparisons, but the shape of the curve is completely meaningless (the data is in no particular sequence). The residuals plot will be useful (actually more so than with dynamic models), but the residual cross-correlations will be useless. When testing the model be sure to use the 'Free running' predictor. Using the other predictors will give misleading results.

Building 'What-If' models

Sometimes it is useful for operators to have models which allow them to forecast the results of particular actions. These 'what-if' models are often used as a half-way house to fully automatic fuzzy control, and allow confidence to be built up amongst the process operators. It should be noted, however, that the control results from 'what-if' supported manual control will be inferior to that

which could be achieved by fully automatic control.

Usually it is necessary to obtain predictions of the output over a 'horizon', rather than at a single point. This can be achieved in CybMod in two ways:

1. The SCADA system can be used to run CybMod as a multi-step ahead predictor. This involves the configuring the SCADA system to repeatedly call CybMod, using previous predictions as inputs for future predictions. This is rather complicated to do, and is not recommended.

2. Multiple models (up to nine) of the process can be created each at a different prediction lag. In this way the effects of a particular change can be seen as a series of 'snapshots' at different times.

As a tutorial, let's build a set of 'what-if' models using method 2.

The data we shall be using in this example ('what-if' on the last disk in the CybOnLine distribution set) was generated by from a simple second order process supplied with two inputs. Both inputs are random, but one changes four times as frequently, and more powerfully, than the other.

The data is arranged in rows of samples and three fields of data. The first data field in the frequently

changing input, the second the other input, and the third is the output.

Load the data into CybMod. Let's assume that we want to build a what-if model based on the first input.

The first thing we have to do is to construct a standard dynamic model to find the minimum lags which can be applied to the process inputs. Using cross-correlation of the raw data and residuals, and trial and error we found the best model of the data (in first-order + dead-time format) to be:

$$Output\,(k)\ '\ R \quad Output\,(k\&1) \quad Input1\,(k\&4)$$

(model built with two reference sets per variable and tested using a free running predictor).

The model above is a standard dynamic model. To convert it into a 'what-if' model we need to make a few changes. In operation, we will know the current value of the output, and will know the value of the input we want to try - in other words we know Output(k) and Input1(k) and want to predict Output(k+?).

From the dynamic model we know that the minimum lag that we can apply to Input1 for the best prediction results is 4. This means that our 'base' what-if model is

$$Output\ (k\%4)\ 'R \quad Output\ (k) \quad Input\ (k)$$

which can be expressed in backward difference form as:

$$Output\ (k)\ 'R \quad Output\ (k\&4) \quad Input1\ (k\&4)$$

We can create a whole series of what-if' models with lags greater than 4 which will allow us to predict more than 4 samples into the future. As we increase our prediction range, however, the accuracy of the 'what-if' models will fall. This is because one, or both, inputs can change within a long prediction period.

Construct the following three sub-models using the data:

$$Output\ (k)\ 'R \quad Output\ (k\&4) \quad Input1\ (k\&4)$$
$$Output\ (k)\ 'R \quad Output\ (k\&8) \quad Input1\ (k\&8)$$
$$Output\ (k)\ 'R \quad Output\ (k\&16) \quad Input1(k\&16)$$

Test the models using a single-step ahead predictor (which is the mode in which these 'what-if' models will be used). You will see that the prediction accuracy falls off with increasing prediction range - at long ranges the prediction may be useless.

Save the model as "c:\program files\cybonline\tut2.pcg" for use in the next chapter.

2

Linking to a SCADA System

CybOnLine is designed to be simple to integrate with your existing SCADA package. This chapter provides you with the general approach which can be used, but the details will be dependent on your particular SCADA software.

Dynamic Data Exchange (DDE)

Dynamic Data Exchange is a method, available in the Microsoft Windows operating systems, for exchanging data between programs. DDE works by using an area of shared memory. It is a bit like using the 'clipboard' to transfer information between programs but occurs completely under program control, without user intervention.

CybOnLine is a *DDE server* which means that it acts primarily as an information source. To connect to CybOnLine your SCADA software must be able to act as a *DDE client*. Consult your SCADA documentation to confirm this.

The key to allowing data to be exchanged through DDE is the *DDE address* of the information to be connected to. The DDE address is made up to three components: the application name; the topic name; and the item name.

The application name is the name of the server which is to be used to supply, or receive, the data. This will always be 'CYBONLINE' for all cases where the SCADA system and CybOnLine are running on the same machine (it is not recommended to run the two pieces of software on different machines over a network).

The topic name is generally used to indicate a particular group of data. In CybOnLine the topic name is set to be the same as the system name specified in CybMod, or "FUZZMOD" if no name was specified.

The item name is the name which connects to a particular piece of data. CybOnLine generates item addresses automatically and lists them against the data field tag names on the CybOnLine faceplate.

Most SCADA systems will allow DDE servers to act

as data I/O sources for a wide variety of input and output variable types (eg analogue outputs, analogue inputs, digital outputs and digital inputs). The usual way to set up a link is to specify DDE as the 'I/O device', and then to include the DDE address as a device I/O address. The method of forming the individual components of the DDE address (application, topic and item) into a single I/O address varies between SCADA systems - consult your SCADA documentation to find out how to do this.

Triggering the calculation

CybOnLine provides you with a choice of ways to trigger a calculation:

Automatic triggering

In this mode CybOnLine keeps track of input information which has been passed from the SCADA package. A model calculation is triggered when a complete set of input data has been received. Once a calculation has been made, all of the input counters are set to zero, and a new, and complete, set of input data has to be sent to trigger a new calculation.

When operating in automatic mode, CybOnLine will generate an error if data arrives out of sequence (i.e. if the same input data is received twice without

an output calculation in-between). This can be useful for warning the user of problems with the DDE connections (either missing or erroneous connections).

Automatic triggering is useful when all the input data is gathered, and modelling calculations are to be carried out, at the same sample interval.

Automatic triggering is the default mode of operation for CybOnLine.

External Triggering

In this mode CybMod only carries out a calculation when it receives an input from a 'trigger' variable. The trigger variable can be an analogue or digital variable, and the value that is sent to CybMod is irrelevant. It is simply the receipt of a value into the trigger address which causes the calculation to be executed.

When operating with an external trigger, CybMod no longer keeps a track of which inputs have been received - it is the responsibility of the SCADA system to ensure that a complete set has been sent.

External triggering can be used in situations where a calculation has to be carried out on demand, rather than at particular intervals (e.g. for operator decision support). It is also useful where model

input data is sampled at different rates, or where a model output is needed less frequently than the sampling rate of the input data.

External triggering is enabled in CybOnLine by starting the program with the '/t' switch on the command line.

Running CybOnLine

CybOnLine must be running before your SCADA system starts to scan. If it isn't running then the SCADA system will try to pick up DDE addresses in CybOnLine which don't exist and scanning errors will result.

There are two ways of ensuring that CybOnLine starts in time:

Adding CybOnLine to the Windows 'StartUp' Folder

Windows includes a special 'StartUp' folder. Programs contained in this folder will automatically be started immediately after the operating system starts. Your SCADA system installation has probably put a start-up component in this folder to automatically start your SCADA software.

To add CybOnLine to this start-up group do the following:

1. Click on the start button, and then point to Settings

2. Click 'Taskbar', and then click the start menu programs tab.

3. Click Add, and then type the following in the command line edit box.

"c:\program files\cybonline" /t "c:\program files\cybonline\model.pcg"

   The directory 'cybonline' may be different if you have chosen to install CybOnLine to a different directory. The '/t' switch instructs CybOnLine to use external triggering. If you want CybOnLine to automatically trigger calculations then simply omit the switch. The 'model.pcg' is the model file that you wish to load into CybOnLine. This model must have been previously created in CybMod, and can be given any file name you fancy.

4. Click the 'Next' button and then click the 'StartUp' folder.

5. Type the name you want to see appear in the Start-up menu and then press the 'Finish' button. The name won't affect the programs operation and is really for your own information.

Multiple copies of CybOnLine can be run

simultaneously. Simply add more start-up command lines to the 'StartUp' folder, remembering to use different model filenames for each.

Adding CybOnLine to the SCADA software's start-up list

Most SCADA packages include the facility of automatically starting additional software on SCADA start-up. The details of how to do this will be in your SCADA packages documentation. Remember that you will need to be able to edit the command line for CybOnLine to allow the model file and trigger switch to be specified.

Again multiple copies of CybOnLine can be run simultaneously by including multiple start-up command lines.

The CybOnLine Faceplate and Linking to the SCADA system.

In this section we shall look at CybOnLine loaded with the Gas tank model you produced in the last chapter.

Start CybOnLine with the gas tank model and automatic triggering by doing the following:

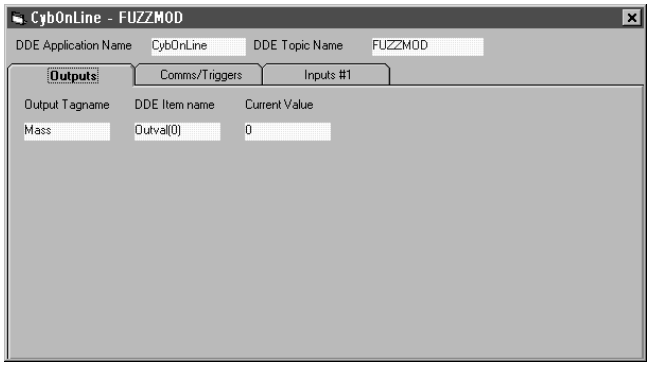
1. Press the Windows 'Start' button and then 'Run'.

2. Press the 'Browse' button and look for the 'cybonline.exe' file. When you find it (it should be in the "c:\program files\cybonline" directory) double click on it.

3. In the edit box in the run dialogue, at the end of the command line add the name of the model file "c:\program files\cybonline\tut.pcg".

4. Press the 'OK' button and CybOnLine should start-up.

The 'Outputs' card



The CybOnLine faceplate is divided into a number of 'cards'. The first of these cards gives details of the model outputs.

At the top of the faceplate the application name and topic name components of the DDE address are given. These address components are common to all the variables in this copy of the program. In this case the topic name is 'FUZZMOD', the default name. This is because we didn't specify a system name when building the model in CybMod. This is not good practice and will give rise to problems if multiple copies of CybOnLine are run. You should always specify unique system names for CybMod models which are to be run in CybOnLine.

On the 'Output' card information is given on the

model outputs included in this model. In this case our model contains only a single sub-model.

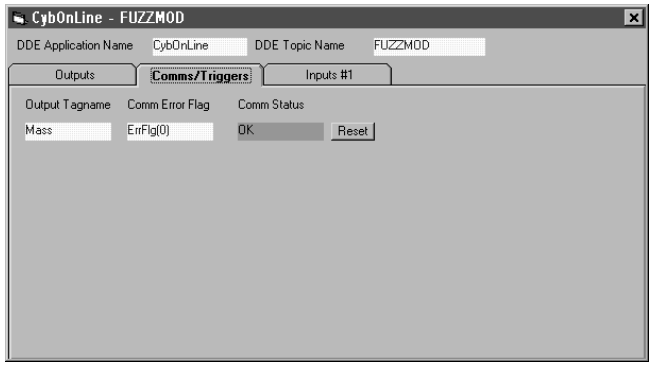
The first column gives the tag name. Where dynamics are present in the model the maximum lag in the sub-model is given in brackets after the output tagname (this is mainly of use in working with 'what-if' models, where the same output may be modelled in several different sub-models). In this case we have a steady-state model (no lags) and nothing is shown after the tag name.

The next column gives the DDE item name to be used for linking this variable with the SCADA system. This item name, along with the application and topic name uniquely identifies this particular output.

The last column on the output card gives the current value of the model outputs. No calculations have yet been carried out and so this value is blank.

Comms/Triggers

Click the 'Comms/Triggers' tab



This card gives information on the status of the DDE communications for each sub-model. When external triggering is selected it will also show details of the trigger address for each sub-model.

The first column on this card, the 'Output Tagname', simply identifies the particular sub-model.

The next column, the 'Comm Error Flag', gives the item name for a digital output which indicates the current communication status of this sub-model. A value of '0' indicates that no problems have been encountered in DDE communications, and a value of '1' means that a problem exists. A DDE problem will fail the sub-model on which it occurs and stop

further calculations pending a reset. The error flag item address should be connected to a SCADA digital input and tied into an alarm or error warning.

The 'Comm Status' column simply gives a visual indication of the current status of the sub-models' links. A communication failure will be shown by a failure message on a red background. Good links are indicated by 'OK' on a green background.

The 'Reset' button is used to reset individual sub-models after a communication failure. With automatic triggering it has to used in-between samples to avoid sequencing problems.

Inputs

Click the 'Inputs #1' tab

There will always one, or more, cards associated with the model inputs. These cards give the variable name, the DDE item name assigned to it by CybOnLine, and the variables current value.

A particular variable can be used as an input for several sub-models. CybOnLine only allocates a single DDE item name for each variable, regardless of the number of times it is used in different sub-models. If a variable is used several times, and automatic triggering is in use, then all the variables in the affected sub-models need to have the same sampling time to avoid sequencing

problems.

SCADA connections

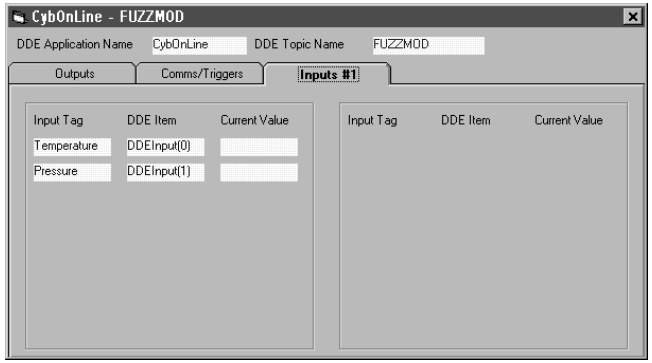
For the model in this example the following SCADA variables/blocks would have to be created and initialised.

An analogue output (temperature) pointing at:

App Name: CybOnLine
Topic: FUZZMOD
Item: DDEInput(0)
Sample time: Same as that at which the data for identification was gathered.

An analogue output (pressure) pointing at:

App Name: CybOnLine
Topic: FUZZMOD



Item: DDEInput(0)

Sample time: Same as that at which the data for identification was gathered.

An Analogue input (Mass) pointing at:

App Name: CybOnLine
Topic: FUZZMOD
Item: Outval(0)
Sample time: Same as that at which the data for identification was gather, but slightly offset from the sample intervals of the two inputs to allow time for processing (5 seconds is plenty).

A Digital input (Communication error alarm) pointing at:

App Name: CybOnLine
Topic: FUZZMOD
Item: ErrFlg(0)
Sample time: Same as that for the other variables.

Linking 'what-if' models

'What-if' models will normally be executed at the request of the operator, which means that CybOnLine has to be set up using the 'external triggering option.

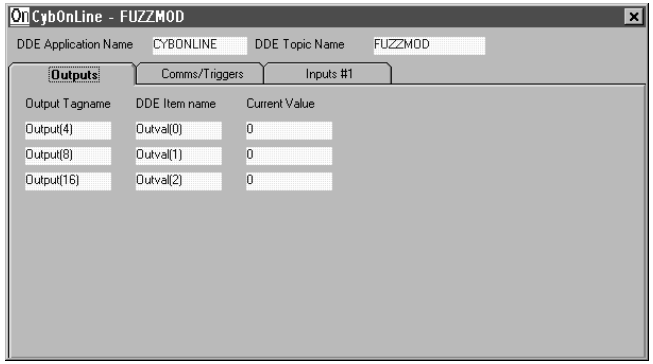
Using the "c:\program files\cybonline\tut2.pcg" (the model file you saved earlier) as an example, the steps in setting up the what-if model are:

1. Run CybOnLine using the external trigger option, e.g.

   cybonline /t tut2.pcg

2. Create two analogue outputs in your SCADA system (one for the current value of the process output, and one for the test value of the input) and connect them to the appropriate DDE addresses in CybMod. The sampling time can be anything you want, and can be a mixture of sampling and exception processing, if this is available on your SCADA system.

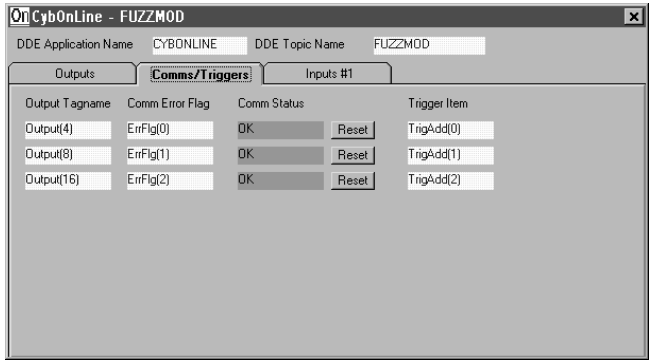
3. Create three analogue inputs and connect them to the three model output DDE addresses.



The three outputs are the outputs of each of the three sub-models you created in CybMod. Output(4) refers to a model where the maximum lag is 4, in this case:

$$Output(k) \text{ ' } R \quad Output(k\&4) \quad Input(k\&4)$$

4. Create three digital inputs and point them to



the error flag variables on the Comms/Triggers

page.

5. Create three trigger outputs and point them at the trigger variable addresses on the Comms/Triggers page. The type of output (digital or analogue) and the value transmitted is irrelevant; the calculation will be triggered when anything is sent to these locations.

6. Add a sequence to your SCADA system which does the following:

   a) The operator presses a button to request a calculation.

   b) The current value of the measurement and trial input are sent to CybOnLine.

   c) Something is transmitted to all three triggers to start a calculation.

   d) Wait for five seconds (this is more than enough time, and in most cases can be reduced)

   e) Acquire the new value of the predicted process outputs from the three CybOnLine outputs.

3

Other Services from Process Cybernetics

We hope that you find CybOnLine a useful and easy to use piece of software. All our software is written in-house and we are continually developing our products. We are very interested in receiving your feedback on our products. If you have any comments to make, or suggestions for improvements to our products, then please get in touch with us at the address given at the end of this chapter.

The rest of this chapter outlines some of the other products and services available from Process Cybernetics.

Other products

CybMod - A fuzzy modelling program for process control

CybMod is the core of our current range of products. It generates relational fuzzy models from process input/output data and saves them in a form suitable for use in CybOnLine and CybFIMC.

CybMod includes tools to analyse and adjust the raw I/O data, and model construction and identification is quick and easy. CybMod also includes a range of tools to evaluate the quality of the models, and to diagnose any problems.

CybFIMC - Model-based fuzzy control

CybFIMC adds fuzzy model-based control capability to your existing Windows based SCADA system. It takes a model, generated in CybMod, and incorporates it into a Fuzzy Internal Model Control (FIMC) scheme. This provides full non-linear feedback control with dead-time compensation, and feed-forward action on any included disturbances.

CybFIMC, like CybOnLine is a DDE server making it really easy to link to existing SCADA installations.

CybFIMC is a MISO (Multi-input/Single-output) controller; it controls a single output variable which can be subjected to several inputs. Multiple copies of CybFIMC can be run simultaneously providing as many loops as you require.

Services from Process Cybernetics

Training

Process Cybernetics offers courses on various aspects of fuzzy modelling and control. These courses include a mixture of both theory and practise.

If you would like to be kept informed of upcoming courses then tick the boxes on your registration card, or contact us directly at the address given at the end of this chapter.

Consultancy

Process Cybernetics can provide consultancy to help support your modelling and control projects. The level of our consultancy involvement can range from telephone support to full project management.

We shall provide a fixed-price quotation for any work we undertake. If you are interested in our consultancy services, then contact us at the address given at the end of the chapter.

Software services

All our software is written in-house, and, as a result, we are able to offer a 'bespoke tailoring' service to make our software suit your particular requirements.

This could involve changing the look of the software, changing the way it operates, or integrating it with other applications.

If you have an application in mind that requires special software, then contact us at the address at the end of the chapter. We will discuss your requirements and provide you with a fixed price quotation for the work.

Chapter 4